



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

## **SISTEM KONTROL POMPA AIR MENGGUNAKAN KONTROLER PID BERBASIS RASPBERRY PI**

Kevin Rosada  
NRP 2212 100 141

Dosen Pembimbing  
Dr. Muhammad Rivai, ST., MT.  
Suwito, ST., MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 141599**

## **WATER PUMP CONTROL SYSTEM USING PID CONTROLLER BASED ON RASPBERRY PI**

Kevin Rosada  
NRP 2212 100 141

Advisor  
Dr. Muhammad Rivai, ST., MT.  
Suwito, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Sistem Kontrol Pompa Air Menggunakan Kontroler PID Berbasis Raspberry Pi**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2017

Kevin Rosada  
NRP. 2212 100 141



# **SISTEM KONTROL POMPA AIR MENGGUNAKAN KONTROLER PID BERBASIS RASPBERRY PI**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Elektronika**

**Jurusan Teknik Elektro**

**Institut Teknologi Sepuluh Nopember**

**Menyetujui:**


**Dosen Pembimbing 1**



**Dr. Muhammad Rivai, ST., MT.**

**NIP. 1969 04 26 1994 03 1 004**

**Dosen Pembimbing 2**



**Suwito, ST., MT.**

**NIP. 1981 10 10 2005 01 1 004**







# **SISTEM KONTROL POMPA AIR MENGGUNAKAN KONTROLER PID BERBASIS RASPBERRY PI**

**Nama** : Kevin Rosada  
**Pembimbing I** : Dr. Muhammad Rivai, ST., MT.  
**Pembimbing II** : Suwito, ST., MT.

## **ABSTRAK**

Distribusi air di Indonesia masih menggunakan metode manual yaitu dengan menggunakan tenaga manusia untuk mengaktifkan dan menonaktifkan pompa distribusi serta menggunakan grafik kebutuhan pelanggan rata – rata sebagai referensinya. Hal ini mengakibatkan distribusi air menjadi tidak beraturan. Oleh karena itu dibutuhkan sistem kontrol pada pompa distribusi yang dapat bekerja secara otomatis menjaga distribusi air sesuai tingkat kebutuhan pelanggan.

Pada tugas akhir ini, kontroler PID (Proporsional, Integral, Derivatif) digunakan untuk menjaga distribusi air dengan mempertahankan debit air keluaran pompa agar tetap stabil pada tingkatan tertentu. Dalam sistem ini kontroler PID akan membandingkan pembacaan tekanan pada pipa dengan nilai tekanan acuan yang telah ditentukan sebelumnya. Hasil penghitungan tersebut digunakan untuk menggerakkan pompa air dengan kecepatan tertentu untuk menyesuaikan tekanan pada pipa. Sistem ini juga memiliki kemampuan proteksi yang akan menonaktifkan pompa ketika berada pada tingkatan tertentu. Hal itu untuk mencegah *overpressure* pada pipa yang dapat menyebabkan kebocoran pada pipa dan kerusakan pada pompa.

Hasil pengujian yang dilakukan pada tugas akhir ini adalah sistem dapat mengontrol debit air keluaran pompa air agar tetap stabil. Sistem diuji dengan mengubah - ubah jumlah kran yang terbuka. Didapatkan hasil sistem yang dibuat dapat menyesuaikan kecepatan pompa air mengikuti jumlah kran yang terbuka dan pengukuran debit air tetap stabil pada debit sekitar 8 Lpm dengan rasio perubahan debit kurang dari 2 Lpm.

Kata kunci : Distribusi Air, Otomatisasi, PID

*Halaman ini sengaja dikosongkan*

## ***WATER PUMP CONTROL SYSTEM USING PID CONTROLLER BASED ON RASPBERRY PI***

***Name*** : Kevin Rosada  
***Supervisor*** : Dr. Muhammad Rivai, ST., MT.  
***Co-Supervisor*** : Suwito, ST., MT.

### ***ABSTRACT***

Water distribution in Indonesia still using manual method which is using human resources to activate and disactivate distribution pump and using a graph of consumer need as a reference. This resulted on a water distribution become unstable. Therefore required control system on distribution pump that can work automatically keeping the water distribution based on customer needs.

In this final project, PID controller (Proportional, Integral, Derivative) is used to keep the water distribution with keeping the water flow of the water pump stable on a certain level. In this system, PID controller will compare the present pressure of the pipe with the reference value that predetermined. The result of that calculation will be used to start the water pump with certain velocity for adjusting the pipe pressure. This system also has an ability for protection that will disactivate the pump when the pressure reach a certain level. It was to prevent overpressure on pipe which can cause leaks on pipe and water pump damage.

The results of tests performed in this final project the system can control the water flow of the water pump stable. System was tested by changing the number of the valve opened. Result obtained that system can adjust the velocity of the water pump following the number of the valve opened and water flow still stable on about 8 Lpm with the ratio of change in water flow are less than 2 Lpm.

Key words : Automation, PID, Water Distributions

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Segala puja dan puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

### **Sistem Kontrol Pompa Air Menggunakan Kontroler PID Berdasarkan Raspberry Pi**

Tugas akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dibuat berdasarkan pada ilmu – ilmu yang telah didapat selama mengikuti perkuliahan, berbagai model literatur penunjang, dan pengarahannya dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak – pihak yang membantu pembuatan tugas akhir ini, khususnya kepada :

1. Bapak Mohamad dan Ibu Betty Feri Astuti yang selalu menjadi motivasi utama penulis dalam menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan Tugas Akhir ini.
3. Bapak Suwito, ST., MT. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan Tugas Akhir ini.
4. Bapak Ir. Hendra Kusuma, M.Eng.sc. , Bapak Ir. Tasripan, MT. , Bapak Ir. Haris Pringadi, MT , dan Bapak Astria Nur Irfansyah, ST., M.Eng sebagai dosen - dosen penguji yang telah memberikan masukan serta saran yang amat berguna dalam pengerjaan Tugas Akhir ini.
5. Intan Puspasari yang telah memberikan waktu dan dukungannya kepada penulis dalam menyelesaikan Tugas Akhir ini.
6. Saudara – saudara yang berada di Divisi Kalpataru yang telah menjadi tempat penulis belajar dan berkumpul bersama.
7. Keluarga besar E52 yang telah menjadi rekan seperjuangan penulis selama melaksanakan kehidupan perkuliahan.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Kritik, saran serta masukan dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak. Penulis juga berharap Tugas Akhir ini dapat dikembangkan lebih lanjut untuk kemajuan industri kedepannya.

Surabaya, 17 Januari 2017

Penulis

## DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	1
1.3. Batasan Masalah .....	2
1.4. Tujuan Penelitian .....	2
1.5. Metodologi Penelitian .....	2
1.6. Sistematika Penulisan.....	3
1.7. Relevansi.....	4
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG .....	5
2.1. Pompa Air Satu Fasa .....	5
2.2. <i>Variable Speed Drive</i> .....	6
2.2.1. Pengertian <i>Variable Speed Drive</i> .....	6
2.2.2. Pengaturan Kecepatan.....	7
2.2.3. <i>Zero Crossing Detector</i> .....	8
2.2.4. <i>TRIODE for Alternating Current (TRIAC)</i> .....	9
2.3. <i>Analog to Digital Converter</i> .....	10
2.3. Sensor Tekanan .....	11
2.4. Raspberry Pi 3 .....	12
2.4.1. Sistem Operasi Raspbian .....	13
2.5. Kontroler PID.....	13
2.5.1. Tuning PID Menggunakan Pendekatan Ziegler-Nichols. ....	15
BAB III PERANCANGAN SISTEM .....	17
3.1. Blok Diagram Sistem .....	17
3.2. Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	18
3.2.1. Perancangan Elektronik .....	18
3.2.1.1. GPIO Raspberry Pi 3 .....	18
3.2.1.2. Perancangan ADC MCP3008.....	19
3.2.1.3. Perancangan <i>Zero Crossing Detector</i> .....	20
3.2.1.4. Perancangan <i>Driver TRIAC</i> .....	20
3.2.1.5. Instalasi Sensor Tekanan .....	21
3.2.2. Perancangan Pipa dan <i>Valve</i> .....	22
3.3. Perancangan Perangkat Lunak .....	22

3.3.1.    Pemrograman Python Pada Raspbian .....	22
3.3.1.1.    Kontroler PID.....	23
BAB IV PENGUJIAN DAN ANALISA .....	25
4.1.    Pengujian ADC MCP3008.....	25
4.2.    Pengujian <i>Zero Crossing Detector</i> .....	26
4.3.    Pengujian Driver TRIAC .....	27
4.4.    Pengujian Sensor Tekanan MP5100GP .....	28
4.5. <i>Tuning</i> Dan Pengujain Respon Kontroler PID .....	29
4.5.1. <i>Tuning</i> Menggunakan Pendekatan Ziegler-Nichols.....	29
4.5.2. <i>Tuning</i> Nilai Kp Menggunakan Metode <i>Trial &amp; Error</i>	31
4.5.3. <i>Tuning</i> Nilai Ki Menggunakan Metode <i>Trial &amp; Error</i>	32
4.5.4. <i>Tuning</i> Nilai Kd Menggunakan Metode <i>Trial &amp; Error</i>	33
4.5.5.    Pengujian Respon Sistem Terhadap Kran Air .....	35
4.6.    Pengujian Pengaruh Sistem Terhadap Debit Air .....	38
BAB V PENUTUP .....	41
5.1.    Kesimpulan .....	41
5.2.    Saran .....	41
DAFTAR PUSTAKA .....	43
LAMPIRAN .....	45
BIODATA PENULIS .....	49



## DAFTAR GAMBAR

<b>Gambar 2.1</b> Konstruksi motor induksi satu fasa .....	6
<b>Gambar 2.2</b> Blok diagram <i>variable speed drive</i> .....	7
<b>Gambar 2.3</b> Karakteristik perubahan kecepatan motor induksi .....	7
<b>Gambar 2.4</b> Grafik tegangan input dan persilangan nol .....	8
<b>Gambar 2.5</b> Gelombang <i>output</i> dan <i>input zero crossing detector</i> .....	8
<b>Gambar 2.6</b> Struktur dan simbol TRIAC .....	9
<b>Gambar 2.7</b> Daerah kerja TRIAC .....	10
<b>Gambar 2.8</b> Karakteristik <i>integral non linearity MCP3008</i> .....	11
<b>Gambar 2.9</b> Konstruksi dan karakteristik sensor MPX5100GP .....	11
<b>Gambar 2.10</b> Raspberry Pi 3 Model B .....	12
<b>Gambar 2.11</b> Tampilan sistem operasi Raspbian .....	13
<b>Gambar 2.12</b> Blok diagram kontroler PID .....	15
<b>Gambar 2.13</b> Parameter pada pendekatan Ziegler Nichols .....	15
<b>Gambar 3.1</b> Blok diagram sistem .....	17
<b>Gambar 3.2</b> Skematik GPIO Raspberry Pi 3 .....	18
<b>Gambar 3.3</b> Skematik ADC MCP3008 .....	19
<b>Gambar 3.4</b> Skematik rangkaian <i>zero crossing detector</i> .....	20
<b>Gambar 3.5</b> Skematik rangkaian <i>driver TRIAC</i> .....	21
<b>Gambar 3.6</b> Skematik rangkaian sensor tekanan MPX5100GP .....	21
<b>Gambar 3.7</b> Perancangan pipa dan <i>valve</i> .....	22
<b>Gambar 3.8</b> <i>Flowchart</i> program .....	23
<b>Gambar 3.9</b> <i>Flowchart</i> kontroler PID .....	24
<b>Gambar 4.1</b> <i>Wiring</i> pengujian ADC MCP3008 .....	25
<b>Gambar 4.2</b> <i>Wiring</i> pengujian <i>zero crossing detector</i> .....	26
<b>Gambar 4.3</b> Hasil pengujian <i>zero crossing detector</i> .....	27
<b>Gambar 4.4</b> <i>Wiring</i> pengujian <i>driver TRIAC</i> .....	27
<b>Gambar 4.5</b> Grafik parameter <i>tuning</i> Ziegler-Nichols pompa air .....	30
<b>Gambar 4.6</b> Hasil <i>tuning</i> PID dengan pendekatan Ziegler-Nichols .....	31
<b>Gambar 4.7</b> Respon sistem setelah ditambahkan kontroler proporsional .....	32
<b>Gambar 4.8</b> Respon sistem setelah ditambahkan kontroler integral ....	33
<b>Gambar 4.9</b> Respon sistem setelah ditambahkan kontroler derivatif ...	34
<b>Gambar 4.10</b> Hasil pengujian pertama ketika kran terbuka dua .....	35
<b>Gambar 4.11</b> Hasil pengujian pertama ketika kran terbuka satu .....	36
<b>Gambar 4.12</b> Hasil pengujian pertama ketika semua kran tertutup .....	36

<b>Gambar 4.13</b>	Hasil pengujian pertama ketika kran dibuka satu .....	37
<b>Gambar 4.14</b>	Hasil pengujian pertama ketika kran dibuka dua.....	38
<b>Gambar 4.15</b>	Hasil pengujian pertama ketika kran dibuka tiga.....	38

## DAFTAR TABEL

<b>Tabel 2.1</b> Konversi Parameter <i>Tuning</i> Ziegler-Nichols.....	16
<b>Tabel 3.1</b> GPIO Pin Raspberry Pi 3 yang digunakan .....	19
<b>Tabel 4.1</b> Hasil pengujian ADC MCP3008.....	26
<b>Tabel 4.2</b> Hasil pengujian <i>driver</i> TRIAC .....	28
<b>Tabel 4.3</b> Hasil pengujian sensor tekanan MPX5100GP.....	29
<b>Tabel 4.4</b> Hasil pengujian nilai $K_p$ .....	31
<b>Tabel 4.5</b> Hasil pengujian nilai $K_i$ .....	32
<b>Tabel 4.6</b> Hasil pengujian nilai $K_d$ .....	34
<b>Tabel 4.7</b> Hasil pengujian pengaruh sistem terhadap debit air .....	39

*Halaman ini sengaja dikosongkan*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Air bersih sudah menjadi kebutuhan pokok yang harus dipenuhi setiap hari oleh masyarakat Indonesia. Faktanya kebutuhan masyarakat Indonesia sekitar 80 – 200 liter setiap orangnya, yang jika dikalkulasikan lagi sekitar 400 – 2000 liter untuk tingkat konsumsi keluarga. Namun karena keterbatasan suplai dan pengguna yang banyak, mengakibatkan tekanan suplai air dari pusat menjadi sangat kecil [1].

Realitanya adalah selama ini distribusi air di Indonesia masih menggunakan cara lama dalam siklus aktivasi pompanya. Distribusi air di Indonesia masih menggunakan grafik siklus konsumsi pengguna. Grafik siklus konsumsi pengguna adalah grafik yang telah dibuat dengan merata – rata penggunaan air konsumen berdasar waktu. Dari grafik siklus konsumsi pengguna tersebut yang dijadikan referensi kapan pompa menyala dan kapan pompa akan mati.

Namun, grafik siklus tersebut tidak selalu menunjukkan keadaan sesungguhnya. Terkadang ketika grafik siklus menunjukkan kebutuhan akan naik pada suatu waktu, realitanya justru sebaliknya. Resikonya ketika pompa air bekerja pada kecepatan maksimum namun tidak ada air yang mengalir menuju konsumen, maka tekanan pada pipa akan semakin besar, ketika tekanan di pipa terus naik hal ini akan dapat merusak pipa dan juga pompa air itu sendiri. Begitu juga ketika grafik siklus menunjukkan kebutuhan akan turun namun realitanya kebutuhan sedang naik, maka aliran air yang mengalir pada konsumen akan kecil akibat dari tekanan pada pipa yang kecil.

Berangkat dari beberapa permasalahan di atas, sistem kontrol pompa air dengan metode PID ini diharapkan bisa menjadi solusi untuk memperbaiki distribusi air bersih di Indonesia agar lebih efisien.

### **1.2. Perumusan Masalah**

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain :

1. Apakah putaran motor pada pompa air dapat berpengaruh pada debit air yang dikeluarkan
2. Jenis sensor apakah yang sesuai untuk mendeteksi tekanan air pada pipa

3. Bagaimana menjaga tekanan pada pipa air agar tetap stabil

### **1.3. Batasan Masalah**

1. Data yang diolah adalah berupa tekanan keluaran dari sensor tekanan MPX5100GP.
2. Sinyal kontrol digunakan untuk mengatur putaran pompa air menggunakan *variable speed drive*.

### **1.4. Tujuan Penelitian**

Adapun tujuan dari penelitian pada tugas akhir ini adalah sebagai berikut :

1. Mengkarakterisasi putaran motor pada pompa air terhadap debit air.
2. Menggunakan sensor MPX5100GP untuk mendeteksi tekanan air pada pipa.
3. Mengimplementasikan sistem kontrol PID untuk membandingkan tekanan pada pipa dengan putaran motor pada pompa air.

### **1.5. Metodologi Penelitian**

Dalam penyelesaian tugas akhir ini digunakan metodologi sebagai berikut :

1. Studi literatur  
Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan tugas akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, *proceeding*, dan artikel-artikel di internet.
2. Perancangan sistem  
Setelah mempelajari literatur yang ada, selanjutnya akan dilakukan perancangan sistem. Perancangan sistem terbagi sebagai berikut :
  - a. Perancangan Perangkat Keras  
Cara kerja dari sistem ini menggunakan sensor tekanan MPX5100GP sebagai parameter untuk mengatur kecepatan pompa air, dan seluruh sistem akan diatur didalam Raspberry Pi.
  - b. Perancangan Variable Speed Drive  
Kecepatan putar motor adalah variabel yang dikontrol pada tugas akhir ini. Karena tegangan yang dikontrol adalah arus jala – jala PLN, maka harus ada *Driver* untuk mengontrol putaran pompa air dengan masukan tegangan DC dan keluaran sinyal kontrol terhadap tegangan jala – jala PLN. Variable Speed Drive terdiri dari rangkaian *Zero Crossing Detector (ZCD)* dan *Triode for Alternating Current (TRIAC)*. Rangkaian ZCD digunakan sebagai penanda tegangan 0

pada tegangan jala – jala PLN yang nantinya dipakai sebagai referensi waktu trigger Raspberry Pi. TRIAC digunakan untuk mengalirkan arus jala – jala PLN ke pompa air dengan *trigger* berupa sinyal DC keluaran dari Raspberry Pi.

c. Perancangan perangkat lunak

Perancangan perangkat lunak pada sistem ini akan dibagi menjadi dua tahapan proses utama, yaitu :

1. Mengkonversikan data sensor tekanan MPX5100GP yang telah didapatkan menjadi data digital.
2. Membuat program kontrol kecepatan pompa air menggunakan kontroler PID dengan mengkarakterisasi tekanan dengan kebutuhan kecepatan pompa air.

d. Pengujian sistem

Pengujian sistem dilakukan dengan beberapa tahapan, yaitu :

1. Menguji kerja pada tiap sistem yang dibuat baik perangkat keras maupun perangkat lunak.
2. Pengujian tiap-tiap bagian pada blok diagram dari rangkaian untuk mengetahui fungsi dan kerja optimal dari sistem yang dibuat.
3. Pengambilan data lapangan sebagai sampel untuk mengetahui ketepatan kerja sistem kontrol pompa air.

e. Pengolahan data

Melakukan analisa data dari hasil eksperimen, sehingga dapat dipilih solusi yang terbaik untuk mengatasi permasalahan yang dihadapi.

f. Penarikan Kesimpulan dan Saran

Penarikan kesimpulan mengacu pada data pengujian, analisis data, dan referensi terkait. Kesimpulan menunjukkan hasil kerja secara garis besar sesuai rumusan masalah yang telah dibuat. Selanjutnya, penarikan saran juga perlu dilakukan sebagai bentuk koreksi terhadap penelitian yang telah dilakukan dan pengembangan penelitian selanjutnya terkait topik serupa.

## **1.6. Sistematika Penulisan**

Laporan tugas akhir ini terdiri dari Lima Bab dengan sistematika penulisan sebagai berikut :

1. BAB 1 PENDAHULUAN

Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi, sistematika penulisan, dan relevansi.

2. BAB 2 DASAR TEORI

Bab ini menjelaskan tentang teori – teori yang dibutuhkan untuk mengerjakan tugas akhir ini, yang meliputi teori dasar tentang sensor MPX5100GP, Raspberry Pi, kontroler PID dan *Variable Speed Drive*.

3. BAB 3 PERANCANGAN SISTEM

Bab ini menjelaskan tentang perancangan perangkat keras dan perangkat lunak sistem secara mendetil. Bab ini juga berisi menjelaskan tentang prosedur pengujian yang dilakukan dalam penelitian.

4. BAB 4 HASIL PENGUJIAN DAN ANALISA DATA

Bab ini menjelaskan tentang hasil pengujian sistem secara keseluruhan dan proses analisisnya.

5. BAB 5 PENUTUP

Bab ini berisi kesimpulan yang meliputi kekurangan-kekurangan pada kerja alat yang didapat dari hasil analisa serta saran untuk pengembangan ke depan.

### **1.7. Relevansi**

Hasil dari tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Sebagai sistem kontrol pompa air yang lebih efisien dari sistem pompa air yang sudah ada.
2. Sebagai dasar penelitian lebih lanjut, agar dapat lebih dikembangkan



## BAB II DASAR TEORI

### 2.1 Pompa Air Satu Phasa

Pompa Air adalah suatu rangkaian elektronika yang dikemas menjadi suatu instrumen, yang mempunyai fungsi sebagai penyedia aliran air dalam debit besar dengan prinsip kerja menghisap air yang tersedia dan mendistribusikan aliran air tersebut kepada setiap saluran keluaran air. Untuk pembagian jenis dari rangkaian pompa air dapat diklasifikasikan melalui kekuatan daya hisap, daya listrik yang dikonsumsi, level ketinggian distribusi air, dan level ketinggian penampungan air. Perhitungan dari daya hisap air dan daya listrik yang dikonsumsi biasanya akan sebanding, jadi apabila pompa air mengalirkan air dalam jumlah debit yang besar maka akan semakin besar pula daya yang dikonsumsi oleh rangkaian pompa air [2].

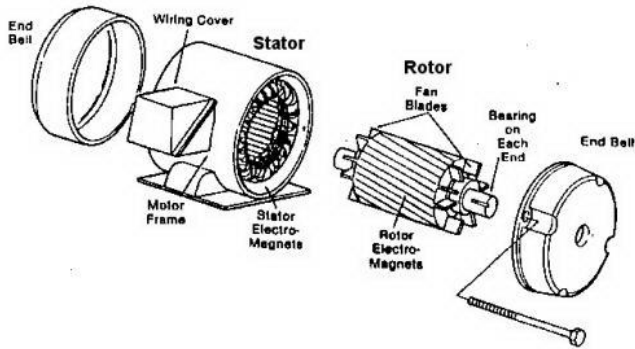
Motor pada pompa air adalah motor induksi 1 phasa. Oleh karenanya rangkaian pompa air akan membutuhkan daya 2 kali lebih besar saat pertama kali dioperasikan, untuk starting motor itu sendiri. Setelah motor sudah aktif maka motor akan memutar baling-baling pompa hingga mencapai putaran yang dibutuhkan oleh pompa air untuk mengkosongkan udara pada ruang pompa, sehingga akan menghasilkan daya hisap kepada sumber air dan memompa kepada saluran-saluran keluaran air. Persamaan kecepatan motor pada pompa air tertulis pada persamaan 2.1.

Stator merupakan bagian yang diam sebagai rangka tempat kumparan stator yang terpasang. Stator terdiri dari : inti stator, kumparan stator, dan alur stator yang bisa dilihat pada gambar 2.1. Motor induksi satu phasa dilengkapi dengan dua kumparan stator yang dipasang terpisah, yaitu kumparan utama (*main winding*) atau sering disebut dengan kumparan berputar dan kumparan bantu (*auxiliary winding*) atau sering disebut dengan kumparan start. Rotor merupakan bagian yang berputar. Pada bagian ini terdiri dari inti rotor kumparan rotor dan alur rotor [3].

$$s = \frac{ns - n}{ns} \quad (2.1)$$

Dimana :

$s$	= slip
$ns$	= kecepatan sinkron [r/min]
$n$	= kecepatan rotor [r/min]



**Gambar 2.1.** Konstruksi Motor Induksi 1 Phasa

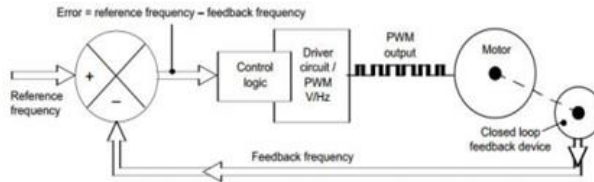
## **2.2 Variable Speed Drive**

### **2.2.1 Pengertian Variable Speed Drive**

Parameter yang dibutuhkan dari motor induksi adalah pengatur kecepatan dan torsi motor. Blok diagram secara umum pada gambar 2.2. Untuk itu dibutuhkan pengaturan yang fleksibel dengan cara mengatur frekuensi tegangan input yang berasal dari tegangan jala – jala PLN. Frekuensi tegangan jala – jala PLN yaitu 50 Hz, dengan menggunakan VSD, frekuensi tersebut dirubah menjadi frekuensi yang diinginkan agar motor dapat berputar pada kecepatan yang diinginkan [4].

Pada umumnya VSD mengubah sumber AC menjadi DC terlebih dahulu. Untuk itu dibutuhkan rangkaian penyearah (rectifier) atau bisa juga menggunakan rangkaian penyearah terkendali (converter). Setelah listrik AC diubah menjadi DC maka diperlukan perataan bentuk gelombang DC yang mengandung ripple dengan menambahkan DC Link atau semacam regulator. Setelah mendapatkan tegangan DC yang baik, langkah berikutnya adalah mengubahnya menjadi listrik AC kembali dengan rangkaian inverter. Rangkaian inverter berfungsi untuk melakukan proses switching secara bergantian terhadap tegangan DC sehingga menghasilkan tegangan AC. Sebelum disambungkan pada motor induksi dibutuhkan rangkaian PWM (Pulse Width Modulation). Rangkaian ini yang akan mencacah listrik DC menjadi listrik AC dengan frekuensi yang diinginkan.

Output dari rangkaian inverter berupa sinyal sinus. Gelombang yang masuk ke motor induksi haruslah berupa tegangan sinus karena bentuk gelombang AC non sinus akan mempengaruhi kualitas daya.



**Gambar 2.2.** Blok diagram Variable Speed Drive

### 2.2.2 Pengaturan Kecepatan

Pengaturan kecepatan motor induksi satu fasa dapat dilakukan dengan mengatur tegangan jala – jala. Semakin besar nilai tegangan pada motor maka kecepatan motor akan semakin besar. Semakin besar beban yang diberikan pada motor mengakibatkan tegangan pada motor akan semakin besar mengikuti batas kecepatan motor. Tegangan pada motor induksi berpengaruh karena perubahan frekuensi mengakibatkan perubahan tegangan yang mengikuti persamaan 2.2.

$$E = 4,44 \cdot f \cdot n \cdot \phi \quad (2.2)$$

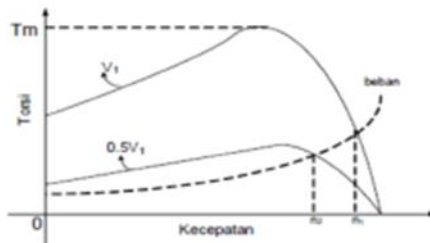
dimana

$E$  : tegangan induksi (V)

$f$  : frekuensi rotor saat berputar (Hz)

$n$  : jumlah belitan-belitan rotor

$\phi$  : fluks maksimum (Wb)

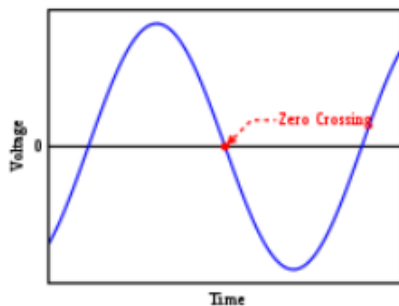


**Gambar 2.3.** Karakteristik Perubahan Kecepatan Motor Induksi

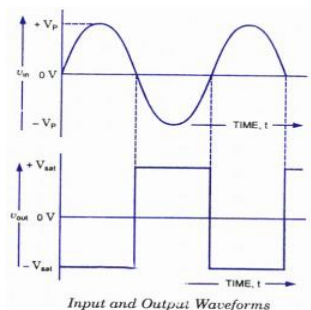
### 2.2.3 Zero Crossing Detector

*Zero crossing detector* adalah rangkaian yang digunakan untuk mendeteksi perpotongan gelombang sinus AC 220 volt saat melewati titik tegangan nol seperti diilustrasikan pada gambar 2.4.

Prinsip kerja *Zero Crossing Detector* adalah dengan membandingkan tegangan referensi yang dihubungkan dengan *ground* (0 Volt). Maka keluarannya adalah gelombang kotak dengan frekuensi sesuai dengan frekuensi AC-nya yaitu 50Hz seperti pada gambar 2.5. Pada saat *Positive Going Transition* (PGT) atau *Negative Going Transition* (NGT) inilah saat terjadi zero. PGT atau NGT inilah yang dibaca oleh mikrokontroller sebagai pendeteksi nol.



**Gambar 2.4.** Grafik Tegangan *Input* dan Persilangan Nol



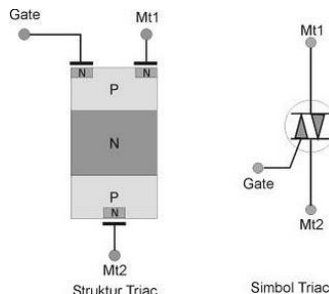
**Gambar 2.5.** Gelombang *Output* dan *Input* Zero Crossing Detector

### 2.2.4 Triode for Alternating Current (TRIAC)

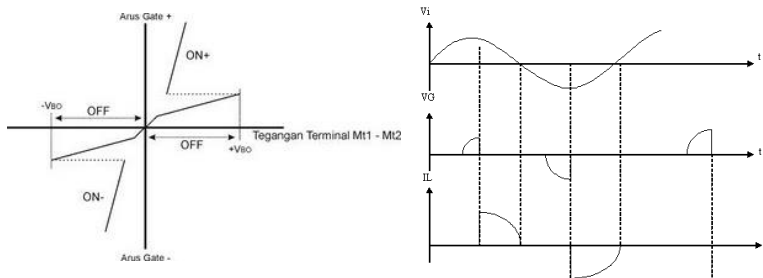
TRIAC, atau *Triode for Alternating Current* adalah sebuah komponen elektronik yang ekuivalen dengan dua SCR yang disambungkan antiparalel dan kaki gerbangnya disambungkan bersama. Nama resmi untuk TRIAC adalah *Bidirectional Triode Thyristor*. Ini menunjukkan saklar dwi arah yang dapat mengalirkan arus listrik ke kedua arah ketika dipicu. Ini dapat disulut baik dengan tegangan positif ataupun negatif pada elektroda gerbang. Sekali disulut, komponen ini akan terus menghantar hingga arus yang mengalir lebih rendah dari arus genggamnya, misal pada akhir paruh siklus dari arus bolak-balik.

TRIAC merupakan komponen 3 elektroda: MT1, MT2, dan *gate* seperti pada gambar 2.6. TRIAC biasanya digunakan pada rangkaian pengendali, penyaklaran, dan rangkian pemacu/*trigger*. Oleh karena aplikasi TRIAC yang demikian luas maka komponen TRIAC biasanya mempunyai dimensi yang besar dan mampu diaplikasikan pada tegangan 100V sampai 800V dengan arus beban dari 0.5A sampai 40A.

Jika terminal MT1 dan MT2 diberi tegangan jala-jala PLN dan *gate* dalam kondisi mengambang maka tidak ada arus yang dilewatkan oleh TRIAC sampai pada tegangan '*break over*' TRIAC tercapai. Kondisi ini dinamakan kondisi *off* TRIAC. Apabila *gate* diberi arus maka tegangan '*break over*' ini akan turun. Semakin besar nilai arus yang masuk ke *gate* maka semakin rendah pula tegangan '*break over*'nya. Kondisi ini dinamakan sebagai kondisi aktif TRIAC. Apabila TRIAC sudah aktif maka TRIAC akan dalam kondisi aktif selama tegangan pada MT1 dan MT2 di atas 0 volt. Apabila tegangan pada MT1 dan MT2 sudah mencapai 0 volt maka kondisi kerja TRIAC akan *off*. Apabila TRIAC sudah menjadi *off*, TRIAC harus di *trigger* kembali seperti pada gambar 2.7.



**Gambar 2.6.** Struktur dan Simbol TRIAC



**Gambar 2.7.** Daerah Kerja TRIAC

### 2.3 Analog to Digital Converter

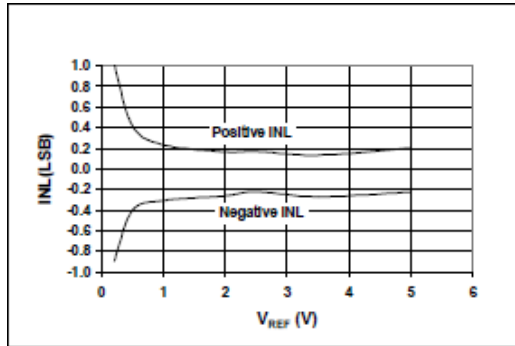
ADC (Analog To Digital Converter) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinyu) menjadi sinyal digital. Perangkat ADC (Analog To Digital Conversion) dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu chip IC.

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi seperti pada persamaan 2.3.

$$\text{Signal} = \frac{\text{Sample}}{\text{Max Value}} \times \text{Reference Voltage} \quad (2.3)$$

ADC (Analog to Digital Converter) memiliki 3 karakter prinsip, yaitu kecepatan sampling, resolusi, dan *Integral Non Linearity*. Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan *sampling* biasanya dinyatakan dalam *sample per second* (SPS). Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Integral Non Linearity adalah tingkat simpangan maksimum antara pembacaan ADC dengan besaran aktualnya [5].

IC MCP3008 merupakan salah satu IC ADC. MCP3008 memiliki resolusi 10 bit dan juga memiliki 8 channel masukan. Selain itu IC ini memiliki kemampuan sampling mencapai 200 ksp/s pada sumber 5V. Bisa dilihat pada gambar 2.8. IC ini memiliki tingkat simpangan pembacaan (INL) yang mendekati 0 [6].

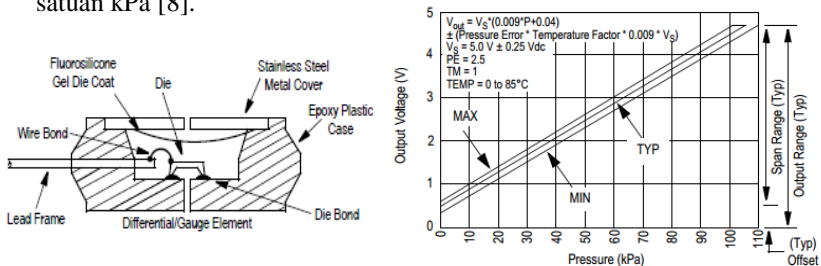


**Gambar 2.8.** Karakteristik *Integral Non Linearity* MCP3008

## 2.4 Sensor Tekanan

Untuk mendeteksi tekanan pada pipa digunakan sensor MPX5100GP. Sensor ini akan menghasilkan sinyal output analog berupa tegangan apabila dideteksi tekanan udara pada tempat tersebut. Tegangan output yang dihasilkan sensor ini berada pada range 0.3 volt – 4.7 volt dengan range tekanan 0 kPa – 100 kPa dan memiliki sensitivitas sebesar 45 mV/kPa [7].

Pada gambar 2.9 di bawah ini menampilkan karakteristik dari sensor MPX5100, semua data yang telah dikumpulkan pada kondisi uji standar. Pada sumbu Y mengindikasikan rasio dari tegangan output ( $V_o$ ) dalam satuan volt, sedangkan sumbu X mendikasikan level tekanan dalam satuan kPa [8].

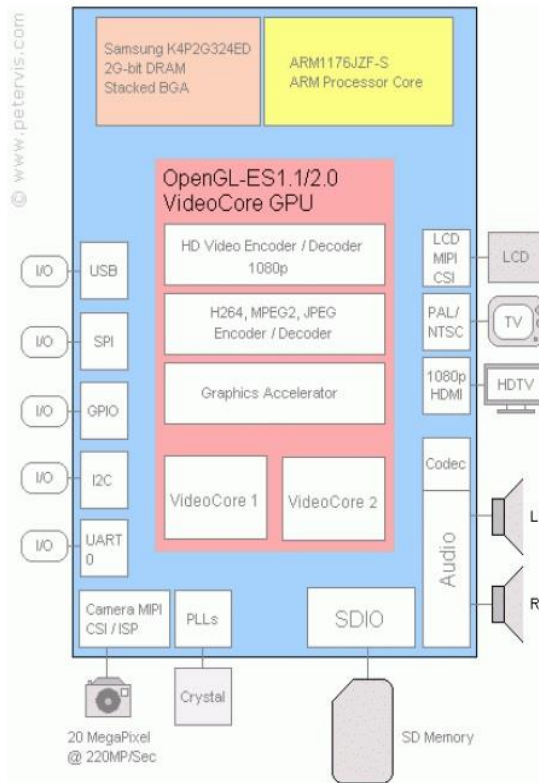


**Gambar 2.9.** Konstruksi dan Karakteristik Sensor MPX5100GP

## 2.5 Raspberry Pi 3

Raspberry Pi (juga dikenal sebagai RasPi) adalah sebuah SBC (Single-Board Computer) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk mengembangkan pengajaran ilmu komputer di sekolah-sekolah [9].

Pada Gambar 2.10 Raspberry Pi menggunakan system on a chip (SoC) dari Broadcom BCM2837, juga sudah termasuk prosesor Quad Cortex A53 1.2 GHz, GPU VideoCore IV dan SDRAM sebesar 1GB (untuk Rev. B) [10].

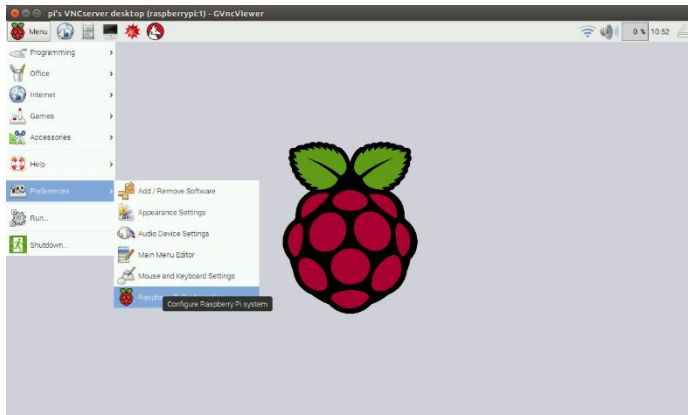


**Gambar 2.10.** Raspberry Pi 3 Model B



### 2.4.1. Sistem Operasi Raspbian

Raspbian adalah sistem operasi gratis yang dikembangkan berdasar pada Linux Debian dan dioptimasi untuk perangkat raspberry pi. Kelebihan dari raspbian adalah banyak tersedianya *library* pendukung perangkat lunak yang dapat didownload secara online dan gratis yang mampu mempermudah penggunaan raspberry pi. Bahasa pemrograman yang digunakan pada Raspbian adalah python.



**Gambar 2.11.** Tampilan Sistem Operasi Raspbian

### 2.6 Kontroler PID

Kontroler PID seperti tergambar pada gambar 2.13 (Proporsional-Integral-Derivatif) merupakan kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industry. Sebuah kontroler PID secara kontinyu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur seperti digambarkan pada gambar 2.13. Kontroler mencoba untuk meminimalkan nilai kesalahan dengan penyetelan variabel kontrol, seperti keran kontrol, damper, atau daya pada elemen pemanas, ke nilai baru seperti pada persamaan 2.4 [11].

$$u(t) = P(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.4)$$

Dengan :

$K_p$  : Gain Proporsional

$K_i$  : Gain Integral

$K_d$  : Gain Derivatif

### 1. Kontroler Proporsional

Kontroler proporsional adalah kontroler yang menghasilkan output yang nilainya berbanding lurus dengan nilai kesalahannya. Kontroler ini memiliki respon yang dapat diatur dengan cara mengalikan kesalahan (error) dengan konstanta  $K_p$ .  $K_p$  disebut konstanta gain proporsional atau gain kontroler. Dalam matematika rumus kontroler proporsional dapat ditulis seperti persamaan 2.5.

$$\begin{aligned} P_{\text{out}} &= K_p e(t) \\ P_{\text{out}} &= K_p (Y_{sp} - Y_m) \end{aligned} \quad (2.5)$$

Ketika gain besar maka menghasilkan output yang besar juga. Namun ketika gain terlalu besar sistem dapat menjadi tidak stabil. Sebaliknya ketika gain kecil maka output juga akan kecil, sehingga membuat waktu respon kontroler menjadi lebih lambat.

### 2. Kontroler Integral

Dalam PID kontroler ini adalah jumlahan error setiap waktu dan mengakumulasi offset yang sebelumnya telah dikoreksi. Error terakumulasi dikalikan dengan  $K_i$  sebagai gain integra dan menjadi keluaran kontroler. Dalam matematika rumus kontroler proporsional dapat ditulis seperti persamaan 2.6.

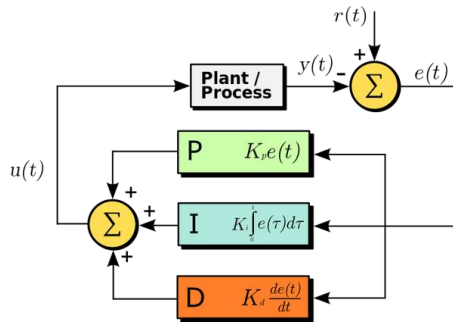
$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau \quad (2.6)$$

Kontroler ini mempercepat perpindahan proses menuju setpoint dan menghilangkan error steady state yang muncul pada kontroler proporsional. Namun, karena kontroler ini merespon error yang telah terakumulasi, maka dapat menyebabkan overshoot

### 3. Kontroler Derivatif

Kontroler derivatif memprediksi perilaku sise dan kemudian memperbaiki waktu dan stabilitas sistem. Turunan error pada proses dihitung dengan menentukan kemiringan error setiap waktu dan mengalikan perubahan tiap waktu dengan gain derivatif  $K_d$ . Dalam matematika rumus kontroler derivatif dapat ditulis seperti persamaan 2.7

$$D_{\text{out}} = K_d \frac{de(t)}{dt} \quad (2.7)$$

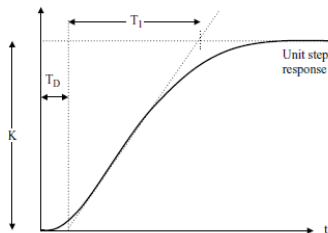


**Gambar 2.12.** Blok Diagram Kontroler PID

### 2.5.1. Tuning PID Menggunakan Pendekatan Ziegler-Nichols

Ketika kontroler PID diterapkan pada sebuah sistem hal yang sangat penting untuk dilakukan adalah melakukan *tuning* untuk mendapatkan respon sistem yang diinginkan. *Tuning* PID dilakukan untuk menemukan nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang bisa menghasilkan respon sistem yang paling bagus. Ada beberapa teknik dalam melakukan *tuning* PID ini, dan yang paling banyak digunakan adalah melalui pendekatan Ziegler-Nichols yang ditemukan oleh J.G Ziegler dan N.B Nichols.

Untuk mendapat nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang harus dilakukan diawal adalah dengan melakukan penentuan parameter  $K$ ,  $T_d$ ,  $T_i$  seperti pada gambar 2.14 dengan cara menarik garis gradien pada grafik respon sistem. Setelah didapatkan nilai parameternya baru dihitung  $K_p$ ,  $K_i$ ,  $K_d$  sesuai Tabel 2.1



**Gambar 2.13.** Parameter pada Pendekatan Ziegler-Nichols

**Tabel 2.1.** Tabel Konversi Parameter *Tuning* Ziegler-Nichols

Controller	$K_p$	$T_i$	$T_d$
Proportional	$\frac{T_1}{KT_D}$		
PI	$\frac{0.9T_1}{KT_D}$	$3.3T_D$	
PID	$\frac{1.2T_1}{KT_D}$	$2T_D$	$0.5T_D$

## BAB III

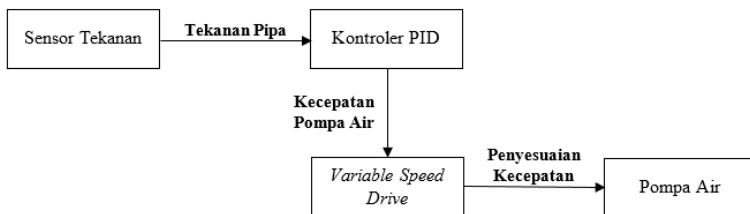
### PERANCANGAN SISTEM

Perancangan sistem pada bab ini terdiri dari dua bagian besar yaitu, perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*).

Perancangan perangkat keras (*hardware*) terdiri atas dua bagian, yaitu meliputi perancangan elektronik dan juga perancangan pipa saluran air pompa. Perancangan elektronik terdiri atas beberapa bagian yaitu, skematik GPIO Raspberry Pi 3, pengendalian motor induksi menggunakan *driver* TRIAC, perancangan ZCD (*Zero Crossing Detector*), perancangan ADC untuk pembacaan sensor menggunakan IC MCP3008, serta perancangan sensor tekanan MPX5100. Sedangkan untuk perancangan pipa, meliputi perancangan percabangan saluran air dan valve.

Untuk perancangan perangkat lunak (*software*) meliputi perancangan pemrograman Python pada Raspberry Pi 3 dan kontroler PID. Perangkat lunak yang dibuat adalah untuk digunakan sebagai pengontrol perangkat keras. Sehingga semua bagian perancangan berhubungan antar satu sama lain.

#### 3.1. Blok Diagram Sistem



**Gambar 3.1.** Blok Diagram Sistem

Pada bab ini dibahas mengenai perancangan alat secara keseluruhan. Dijelaskan pula lebih terperinci tiap – tiap bagian atau blok – blok penyusun alat ini berupa blok diagram sistem seperti tergambar pada gambar 3.1.

Sensor Tekanan akan menghasilkan data berupa hasil pengukuran tekanan pada pipa. Data tersebut kemudian diolah dengan kontroler PID

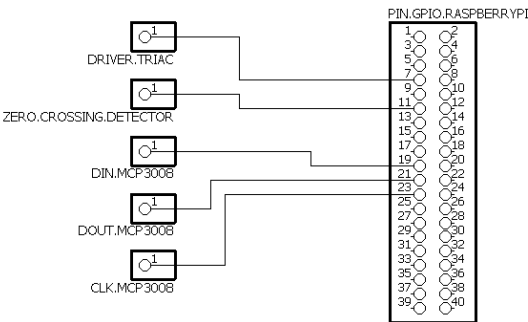
untuk mendapatkan kecepatan pompa air untuk menyesuaikan tekanan pada pipa agar tetap stabil pada besaran tertentu. Data yang telah diolah tersebut kemudian akan digunakan untuk mengendalikan kecepatan pompa air dengan menggunakan *Variable Speed Drive*, sistem mempercepat dan memperlambat kecepatan pompa air untuk menyesuaikan tekanan pada pipa.

### 3.2. Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras (*hardware*) terdiri atas dua bagian, yaitu meliputi perancangan elektronik dan juga perancangan pipa saluran air pompa. Perancangan elektronik terdiri atas beberapa bagian yaitu, skematik GPIO Raspberry Pi 3, pengendalian motor induksi menggunakan *driver* TRIAC, perancangan ZCD (*Zero Crossing Detector*), perancangan ADC untuk pembacaan sensor menggunakan IC MCP3008, serta perancangan sensor tekanan MPX5100. Selanjutnya untuk perancangan pipa, meliputi perancangan percabangan saluran air dan pemasangan valve.

#### 3.2.1. Perancangan Elektronik

##### 3.2.1.1. GPIO Raspberry Pi 3



**Gambar 3.2.** Skematik GPIO Raspberry Pi 3

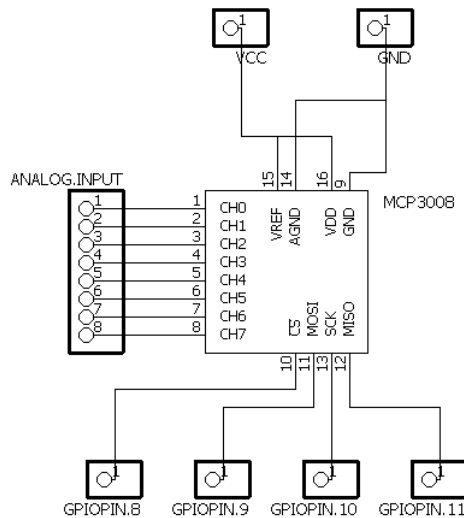
Dari Gambar 3.2 skematik GPIO Raspberry Pi 3 dapat diketahui mengenai sambungan pin – pin pada Raspberry Pi 3. Pin – pin Raspberry Pi 3 yang digunakan pada tugas akhir dituliskan pada tabel 3.1.

**Tabel 3.1.** GPIO Pin Raspberry Pi 3 yang Digunakan

BCM PIN	FUNGSI
4	Trigger untuk driver TRIAC
17	Input sinyal <i>Zero Crossing Detector</i>
8,9,10,11	Membaca hasil olahan ADC MCP3008

### 3.2.1.2. Perancangan ADC MCP3008

Analog to Digital Converter (ADC) digunakan untuk mengubah suatu data analog menjadi data digital agar sinyal masukan analog dapat diproses oleh Raspberry Pi 3. Adanya perancangan ADC ini dikarenakan Raspberry Pi 3 tidak memiliki pin ADC. Raspberry Pi 3 hanya dapat menerima sinyal digital saja. Proses pembacaan ADC MCP3008 menggunakan komunikasi SPI (*Serial Peripheral Interface*). SPI merupakan salah satu protokol komunikasi data sinkron antara dua piranti digital atau lebih. Salah satu piranti berperan sebagai Master sedangkan piranti lain berperan sebagai Slave. Raspberry Pi 3 digunakan sebagai master dan MCP3008 sebagai slave. Skematik ADC MCP3008 seperti pada gambar 3.3.

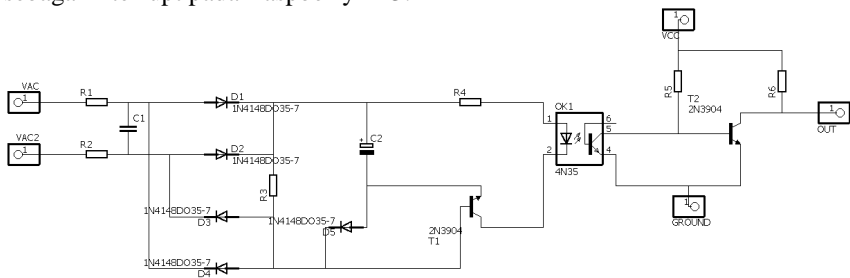


**Gambar 3.3.** Skematik ADC MCP3008

### 3.2.1.3. Perancangan Zero Crossing Detector

*Zero Crossing Detector* adalah rangkaian yang digunakan untuk mendeteksi persimpangan titik nol pada gelombang sinus tegangan jala – jala AC 220V. Rangkaian *Zero Crossing Detector* pada tugas akhir ini menggunakan optocoupler IC 4N35. Skematik rangkaian *Zero Crossing Detector* seperti pada gambar 3.4.

Pada rangkaian ZCD tersebut digunakan prinsip *switching* pada empat diode yang terhubung pada sumber. Pada saat fasa positif dioda yang aktif adalah dioda D1 dan D3 sehingga arus akan mengalir dan mengisi kapasitor C2, ketika tegangan mencapai fasa nol, tidak ada arus yang mengalir pada diode maka kapasitor akan *discharge* dan mengalirkan arus menuju optocoupler sehingga memberikan logika satu. Sama halnya ketika fasa negatif yang membedakan adalah dioda yang aktif adalah D2 dan D4. Output dari rangkaian ZCD tersebut akan masuk sebagai interrupt pada Raspberry Pi 3.



**Gambar 3.4.** Skematik Rangkaian Zero Crossing Detector

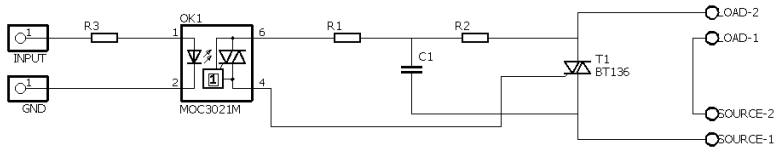
### 3.2.1.4. Perancangan Driver TRIAC

*Driver TRIAC* digunakan untuk mengatur tegangan yang masuk ke motor dengan metode *switching*. *Switching TRIAC* Raspberry Pi 3 menggunakan tegangan 3.3 VDC. Sedangkan tegangan yang masuk ke pompa ait adalah tegangan jala – jala PLN 220-230 VAC. Oleh karena itu pada rangkaian *driver* ini dibutuhkan suatu pemisah antara tegangan rangkaian kontrol yang berupa tegangan DC terhadap rangkaian daya yang berupa tegangan jala jala AC. Rangkaian *driver TRIAC* AC 1 Fasa seperti pada gambar 3.5.

Pada rangkaian *driver* tersebut digunakan MOC3021 sebagai pengaman antara tegangan DC dengan tegangan tinggi AC. TRIAC BTA16 digunakan sebagai saklar antara sumber tegangan dan beban



pompa air. Rangkaian tersebut sesuai dengan yang ada pada datasheet IC MOC3021 dengan beban induktif.. Arus yang mengalir pada gate TRIAC kurang lebih 15 – 50 mA.

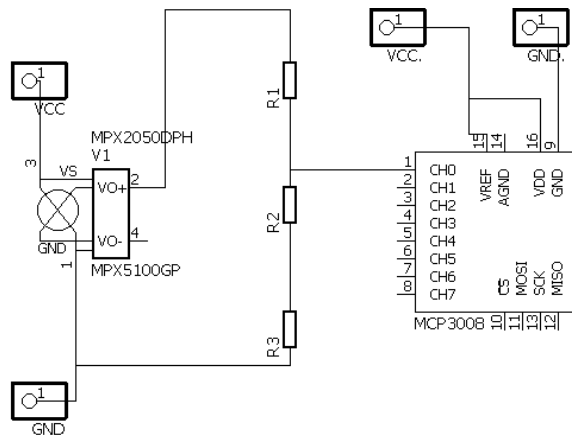


**Gambar 3.5.** Skematik Rangkaian *Driver Triac*

### 3.2.1.5. Instalasi Sensor Tekanan

Pada tugas akhir ini digunakan sensor tekanan MPX5100GP. Sensor ini digunakan untuk membaca tekanan pada pipa. MPX5100GP ini dapat mendeteksi tekanan hingga 100 kPa dan dapat mendeteksi tekanan dari air maupun udara.

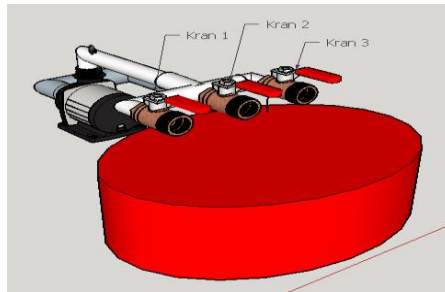
Namun sensor ini membutuhkan sumber sebesar 5V sedangkan pada Raspberry Pi 3 semua sistemnya menggunakan sumber 3.3V, Maka dibutuhkan rangkaian pembagi tegangan 5V menjadi 3.3V. Skematik rangkaian sensor tekanan MPX5100GP seperti pada gambar 3.6.



**Gambar 3.6.** Skematik Rangkaian Sensor Tekanan MPX5100GP

### 3.2.2. Perancangan Pipa dan Valve

Perancangan pipa dan *valve* disini maksudnya adalah untuk mengatur jalannya air dari inlet pompa air, menuju ke outlet pompa air, sampai keluar melalui *valve*. Pada inlet dan outlet pompa air diameter pipa berdiameter  $\frac{3}{4}$ " sedangkan pada *valve* diameternya menjadi  $\frac{1}{2}$ ", oleh karenanya sebelum menuju *valve* harus diberi reducer terlebih dahulu. Perancangan pipa seperti pada gambar 3.7.



Gambar 3.7. Perancangan Pipa dan Valve

### 3.3. Perancangan Perangkat Lunak (*Software*)

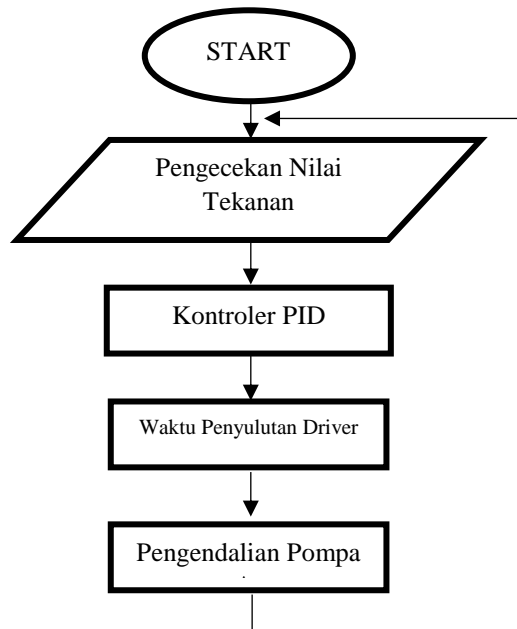
Perancangan perangkat lunak (*software*) pada Tugas Akhir ini terdiri dari beberapa pemrograman Python.

#### 3.3.1. Pemrograman Python pada Raspbian

Python adalah bahasa pemrograman yang bersifat *open-source*, yang memiliki banyak *library* yang mudah diakses dan dapat digunakan untuk berbagai kebutuhan pemrograman. Tata Bahasa yang digunakan pada Pythonpun bisa dikatakan mudah. Selain itu bahasa pemrograman Python ini memiliki aturan *layout source code* yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang *source code*. Pada Raspbian, Python dapat digunakan untuk mengontrol GPIO pada Raspberry Pi. Dengan mengimport *library* GPIO dari server pypi dan beberapa *library* pendukung lainnya GPIO dari Raspberry Pi dapat diakses untuk berbagai macam kebutuhan. *Flowchart* program keseluruhan seperti gambar 3.8.

Ketika program dieksekusi, program akan langsung membaca besaran tekanan pada pipa. Besaran tekanan tadi akan diolah dan dibandingkan oleh kontroler PID untuk menghasilkan sinyal kontrol. Sinyal kontrol tersebut berupa waktu penyulutan driver TRIAC. Driver TRIAC akan menyesuaikan kecepatan pompa air sesuai dengan waktu

penyulutan yang diberikan oleh kontroler PID. Siklus ini akan berlangsung secara terus – menerus.



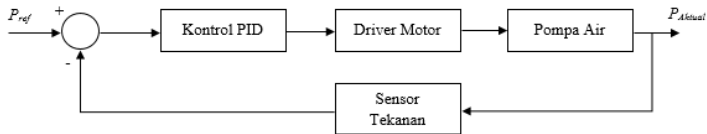
**Gambar 3.8.** *Flowchart Program*

#### **3.3.1.1. Kontroler PID**

Sistem kontrol PID terdiri dari tiga jenis cara pengaturan yaitu kontrol P (*Proportional*), I (*Integral*), dan D (*Derivative*). Dalam penerapannya masing – masing jenis cara pengaturan tersebut dapat bekerja sendiri – sendiri ataupun gabungan. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter  $K_p$ ,  $K_i$ , dan  $K_d$ . Flowchart program PID pada Python seperti pada gambar 3.9.

Kontroler PID digunakan untuk menyesuaikan kecepatan pompa air untuk menjaga tekanan agar tetap stabil pada satu besaran tekanan tertentu. Setelah mendapat data besaran tekanan hasil pembacaan sensor, kontroler PID akan menghitung *error* antara pembacaan tekanan sensor dengan *setpoint*. Setelah *error* diproses, kontroler PID akan menghasilkan

sinyal kontrol berupa waktu penyulutan *driver* motor. *Driver* akan menyesuaikan kecepatan pompa air.



**Gambar 3.9** Flowchart Kontroler PID

## BAB IV

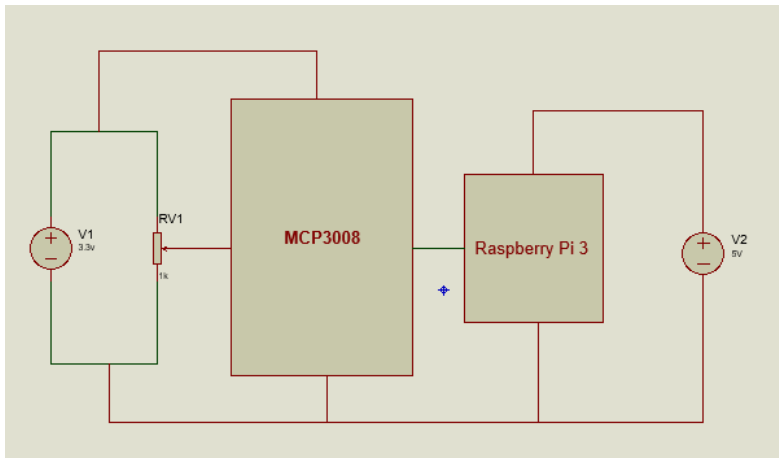
### HASIL PENGUJIAN DAN ANALISA DATA

Pada bab ini akan dibahas mengenai hasil pengujian dan analisa data dari sistem kontrol pompa air yang telah dibuat sebelumnya. Pengujian alat ini bertujuan untuk memastikan agar peralatan yang dibuat berfungsi dengan baik dan sesuai teori yang telah dipelajari sebelumnya.

Pengujian ini terdiri dari pengujian ADC MCP3008, pengujian *zero crossing detector*, pengujian *driver TRIAC*, pengujian sensor tekanan MPX5100GP, *tuning* dan pengujian respon kontroler PID, serta pengujian pengaruh kontroler terhadap debit air. Setelah melakukan beberapa pengujian pada alat yang telah dibuat, data yang diperoleh akan dianalisa untuk mengetahui proses kerja dari seluruh sistem alat yang dibuat.

#### 4.1. Pengujian ADC MCP3008

Pengujian ADC MCP3008 dilakukan untuk mendapat data ukur yang valid antara *input* dan *output*. Pengujian ADC MCP3008 adalah dengan mengatur tegangan input sebesar 0V – 3.3V menggunakan potensiometer seperti pada gambar 4.1 Hasil output ditampilkan pada monitor Python Shell. Hasil pengujian ADC MCP3008 tertulis pada tabel 4.1.



**Gambar 4.1.** Wiring Pengujian ADC MCP3008

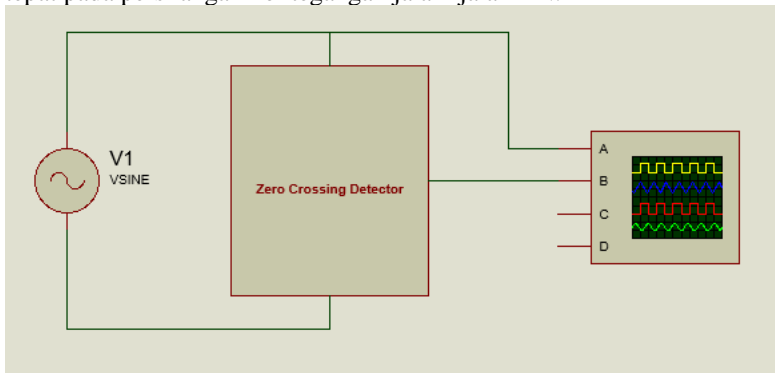
**Tabel 4.1.** Hasil Pengujian ADC MCP3008

<i>Input</i>	<i>Output</i>
0 V	0
0.5 V	155
1.0 V	311
1.5 V	465
2.0 V	620
2.5 V	776
3.0 V	931
3.3 V	1023

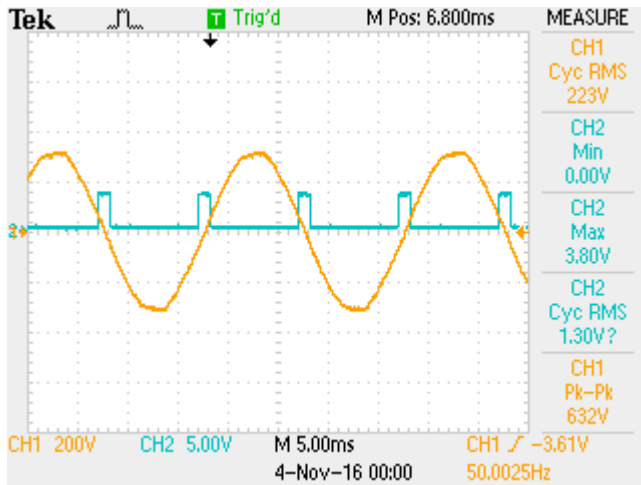
#### 4.2. Pengujian *Zero Crossing Detector*

Pengujian *zero crossing detector* bertujuan untuk mengetahui apakah rangkaian ZCD yang telah dibuat telah menghasilkan sinyal tepat pada persilangan nol tegangan jala – jala PLN. Pengujian ini dilakukan dengan cara membandingkan sinyal hasil keluaran *zero crossing detector* dengan sinyal AC tegangan jala – jala PLN. *Wiring zero crossing detector* seperti pada gambar 4.2.

Setelah dilakukan pengujian pada rangkaian zero crossing detector didapatkan hasil perbandingan sinyal tegangan AC dari tegangan jala – jala PLN dengan hasil output pada rangkain seperti pada gambar 4.3. Dari hasil pengujian bisa didapatkan sinyal output zero crossing detector telah tepat pada persilangan nol tegangan jala – jala PLN.



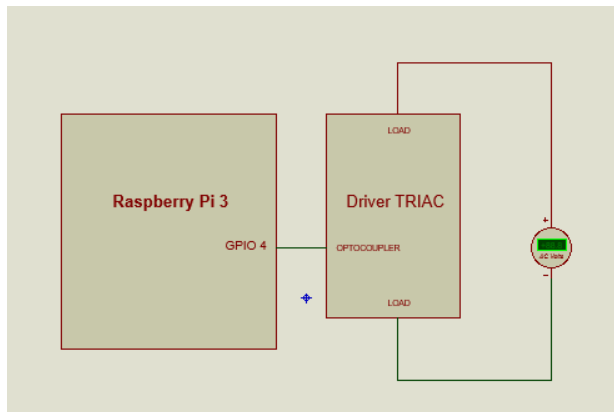
**Gambar 4.2.** Wiring Pengujian *Zero Crossing Detector*



**Gambar 4.3.** Hasil Pengujian Zero Crossing Detector

#### 4.3. Pengujian Driver TRIAC

Tujuan pengujian *driver* TRIAC adalah untuk mengetahui tegangan efektif output yang di *drive* oleh TRIAC untuk mengontrol kecepatan motor pada pompa air. Setelah mendapat sinyal dari *zero crossing detector*, Raspberry Pi akan memberikan sinyal penyulutan TRIAC pada waktu – waktu tertentu seperti pada gambar 4.4. Hasil pengujian Driver TRIAC tetulis pada tabel 4.2.



**Gambar 4.4.** Wiring Pengujian Driver TRIAC

**Tabel 4.2.** Hasil Pengujian *Driver TRIAC*

Waktu Penyulutan (ms)	<i>Output</i> (Volt)
0	226
1	223
2	220
3	209
4	164
5	122
6	73
7	55
8	38
> 9	2

#### **4.4. Pengujian Sensor Tekanan MPX5100GP**

Pengujian sensor tekanan MPX5100GP bertujuan untuk mengkarakterisasi perubahan tekanan pada pipa akibat dari perubahan terbuka dan tertutupnya kran air. Dari karakterisasi tersebut nantinya dapat ditentukan kemampuan maksimal kerja pompa air dan juga batas akhir tekanan pipa terbesar untuk sebagai proteksi terhadap kebocoran pipa dan kerusakan motor pada pompa air.

Pengujian ini dilakukan dengan cara mengaktifkan pompa tanpa menggunakan kontroler kecepatan, lalu menutup kran satu persatu. Nantinya akan didapat pembacaan tekanan oleh MPX5100GP dengan besaran kPa.

Setelah dilakukan pengujian sensor tekanan MPX5100GP didapatkan hasil seperti pada tabel 4.3. Sebagai catatan pengukuran tekanan air pada pipa sangat mudah berubah. Hasil pengukuran tekanan pada tabel 4.3 dihitung dengan menghitung rata – rata hasil yang ditunjukkan saat pembacaan ADC.



**Tabel 4.3.** Hasil Pengujian Sensor Tekanan MPX5100GP

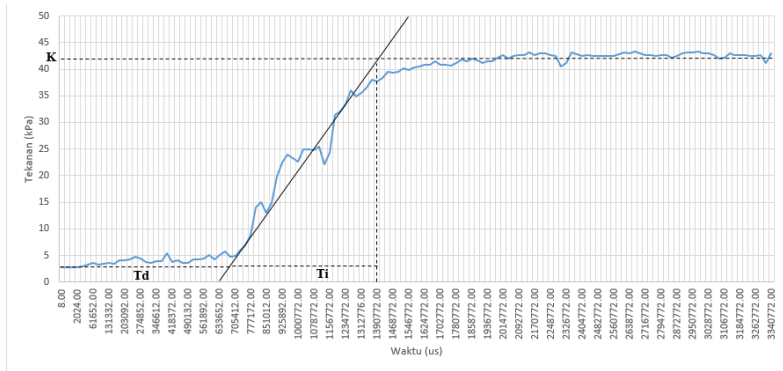
Kran 1	Kran 2	Kran 3	Tekanan (kPa)
Tutup	Tutup	Tutup	97.57
Tutup	Tutup	Buka	41.57
Tutup	Buka	Tutup	43.30
Tutup	Buka	Buka	19.57
Buka	Tutup	Tutup	38.44
Buka	Tutup	Buka	18.41
Buka	Buka	Tutup	18.80
Buka	Buka	Buka	11.59

#### **4.5. *Tuning* dan Pengujian Respon Kontroler PID**

Di dalam pengujian ini, dicari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  dicari agar pergerakan pompa air beresilasi seminimum mungkin ketika diberi suatu nilai *setpoint*. Nilai *setpoint* berupa nilai tekanan yang sudah ditentukan pada program Raspberry Pi .

##### **4.5.1. *Tuning* Menggunakan Pendekatan Ziegler-Nichols**

Di dalam pengujian ini dicari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  menggunakan pendekatan Ziegler-Nichols. Grafik respon pompa yang digunakan untuk penentuan parameter adalah grafik respon pompa ketika diaktifkan dari kondisi mati pada saat kondisi kran hanya terbuka satu. Sumbu x pada grafik merupakan waktu dalam us. Sedangkan sumbu y adalah tekanan dalam kPa. Penentuan parameter pendekatan Ziegler-Nichols seperti pada gambar 4.5.



**Gambar 4.5.** Grafik Parameter *Tuning* Ziegler-Nichols Pompa Air

$$K = 39.64579 \text{ kPa} = 39645.79 \text{ Pa}$$

$$Ti = 1390772 - 633652 = 757120 \text{ us} = 0.757120 \text{ s}$$

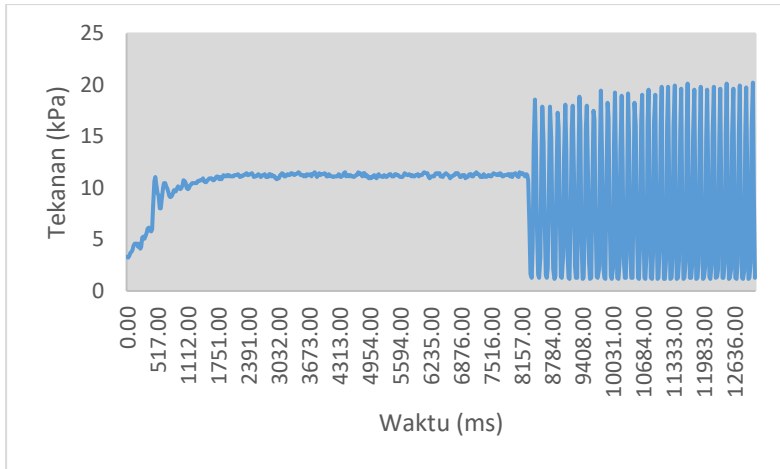
$$Td = 633652 - 8 = 633644 \text{ us} = 0.633644 \text{ s}$$

$$Kp = \frac{1.2Ti}{KTd} = \frac{1.2 \times 0.757120}{39645.79 \times 0.633644} = 0.000036$$

$$Ki = 2Td = 2 \times 0.633644 = 1.267288$$

$$Kd = 0.5Td = 0.5 \times 0.633644 = 0.316822$$

Hasil *Tuning* PID menggunakan pendekatan Ziegler-Nichols mendapatkan hasil  $Kp$  sebesar 0.000036,  $Ki$  sebesar 1.267288 dan  $Kd$  sebesar 0.316822. Setelah nilai  $Kp$ ,  $Ki$ ,  $Kd$  tersebut dimasukkan ke dalam program didapati respon sistem seperti pada gambar 4.6. Pada gambar 4.6 menunjukkan sistem yang tidak stabil. Pompa bergetar tidak beraturan akibat respon pompa yang lambat dan perubahan tekanan pada pipa yang sangat cepat sehingga terjadi ketidakstabilan pada sistem. Karena hasil respon sistem yang tidak stabil sehingga hasil *tuning* menggunakan pendekatan Ziegler-Nichols tidak dapat digunakan.



**Gambar 4.6.** Hasil *Tuning* PID dengan Pendekatan Ziegler-Nichols

#### 4.5.2. *Tuning* Nilai Kp Menggunakan Metode *Trial & Error*

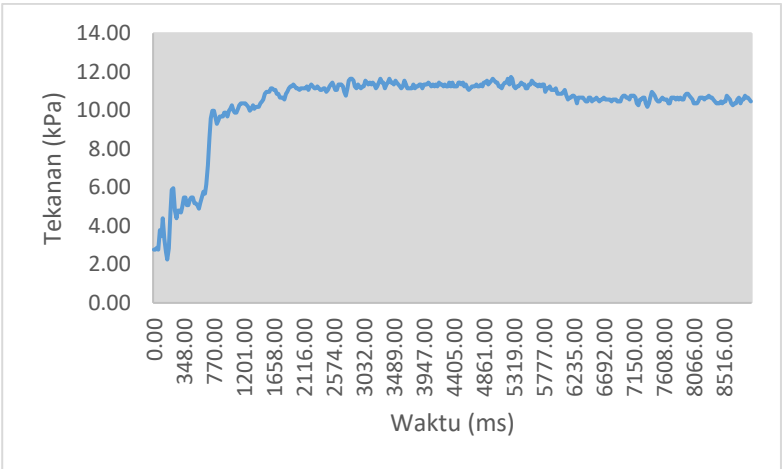
Pengujian ini dilakukan untuk melihat besaran tekanan terhadap respon motor yang telah dikontrol oleh kontroler PID dalam mencapai nilai referensi atau *setpoint*. Penyesuaian kecepatan motor menggunakan kontroler PID dengan pengujian berulang – ulang atau *trial & error* untuk mendapatkan nilai Kp pada tekanan referensi 11 kPa.

**Tabel 4.4.** Hasil Pengujian Nilai Kp

Kp	Ki	Kd	<i>Setpoint</i> (kPa)	<i>Rise</i> <i>Time</i>	<i>Settling</i> <i>Time</i>
0.0007	0	0	11	1s	>10s
0.00001	0	0	11	1s	5s
0.000024	0	0	11	1s	5s

Dari pengujian Kp pada tabel 4.4 didapat pada Kp sebesar 0.0007 sistem tidak stabil tekanan sangat fluktuatif akibat dari respon motor yang fluktuatif juga. Pada Kp sebesar 0.00001 didapat respon yang lebih baik dengan rata – rata tekanan sebesar 10,52 kPa. Pada gambar 4.7 didapat respon yang stabil dan lebih baik lagi dengan rata – rata tekanan sebesar 10.61 kPa. Maka dari hasil pengujian dipilih Kp sebesar 0.000024 sebagai

nilai  $K_p$  sistem karena memiliki tingkat kesalahan yang lebih kecil dibandingkan nilai  $K_p$  yang lain dan memiliki respon waktu yang lebih cepat.



**Gambar 4.7.** Hasil Respon Sistem Setelah Ditambahkan Kontroler Proporsional

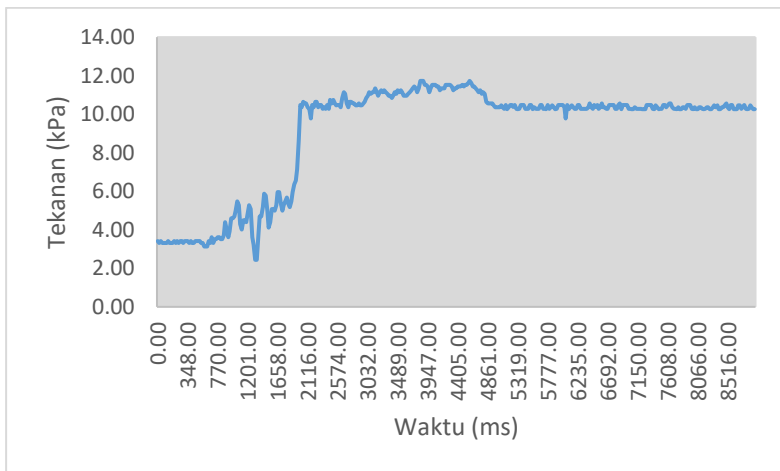
**4.5.3. Tuning Nilai  $K_i$  Menggunakan Metode *Trial & Error***

Pengujian ini dilakukan untuk melihat besaran tekanan terhadap respon motor yang telah dikontrol oleh kontroler PID dalam mencapai nilai referensi atau *setpoint*. Penyesuaian kecepatan motor menggunakan kontroler PID dengan pengujian berulang – ulang atau *trial & error* untuk mendapatkan nilai  $K_i$  pada tekanan referensi 11 kPa.

**Tabel 4.5.** Hasil Pengujian Nilai  $K_i$

$K_p$	$K_i$	$K_d$	<i>Setpoint (kPa)</i>	<i>Rise Time</i>	<i>Settling Time</i>
0.000024	0.00000004	0	11	<1s	5s
0.000024	0.000000007	0	11	<1s	5s
0.000024	0.0000000001	0	11	<1s	4.5s

Dari pengujian  $K_i$  pada tabel 4.5 didapat pada  $K_i$  dengan nilai 0.0000004 pompa bergetar tidak beraturam akibat overshoot yang sangat besar. Pada nilai  $K_i$  sebesar 0.00000007 didapat respon yang lebih baik namun diawali dengan overshoot yang sangat besar juga namun dengan waktu yang lebih singkat. Pada nilai  $K_i$  sebesar 0.000000001 pada gambar 4.8 didapat respon yang stabil dan lebih baik lagi dengan rata – rata tekanan sebesar 10.37 kPa. Maka dari hasil pengujian dipilih  $K_i$  sebesar 0.00000000001 sebagai nilai  $K_i$  sistem karena memiliki tingkat kesalahan yang lebih kecil dibandingkan nilai  $K_i$  yang lain dan memiliki respon waktu yang lebih cepat.



**Gambar 4.8.** Hasil Respon Sistem Setelah Ditambahkan Kontroler Integral

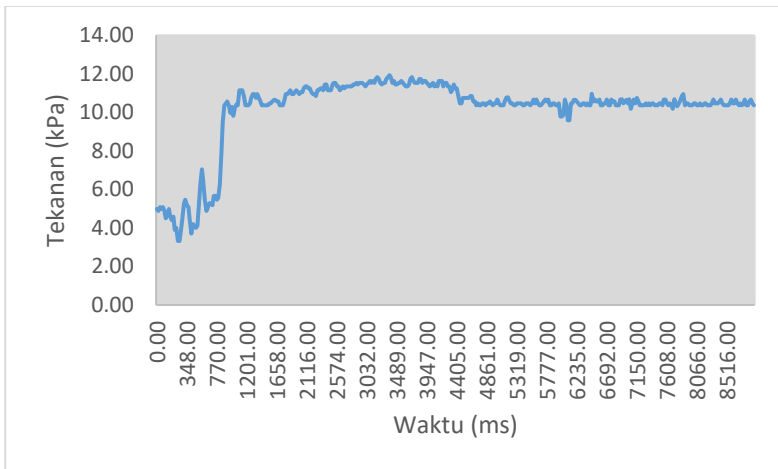
#### 4.5.4. *Tuning Nilai $K_d$ Menggunakan Metode Trial & Error*

Pengujian ini dilakukan untuk melihat besaran tekanan terhadap respon motor yang telah dikontrol oleh kontroler PID dalam mencapai nilai referensi atau *setpoint*. Penyesuaian kecepatan motor menggunakan kontroler PID dengan pengujian berulang – ulang atau *trial & error* untuk mendapatkan nilai  $K_d$  pada tekanan referensi 11 kPa.

**Tabel 4.6.** Hasil Pengujian Nilai Kd

Kp	Ki	Kd	Setpoint (kPa)	Rise Time	Settling Time
0.000024	0.000000001	0.06	11	<1s	>10s
0.000024	0.000000001	0.003	11	<1s	4.5s
0.000024	0.000000001	0.0009	11	<1s	4.2s

Dari pengujian Kd pada tabel 4.6 didapat pada gambar 4.9 dengan Kd 0.06 sistem mendapat overshoot yang sangat besar sehingga mengakibatkan pompa bergetar tidak beraturan. Pada nilai Kd sebesar 0.003 didapat respon yang lebih baik dengan respon waktu yang lebih cepat juga. Pada gambar 4.7 didapat respon yang stabil dan lebih baik lagi dengan rata – rata tekanan sebesar 10.47 kPa. Maka dari hasil pengujian dipilih Kd sebesar 0.0009 sebagai nilai Kd sistem karena memiliki tingkat rasio error yang lebih kecil dibandingkan nilai Kd yang lain dan memiliki waktu respon yang lebih cepat.



**Gambar 4.9.** Hasil Respon Sistem Setelah Ditambahkan Kontroler Derivatif

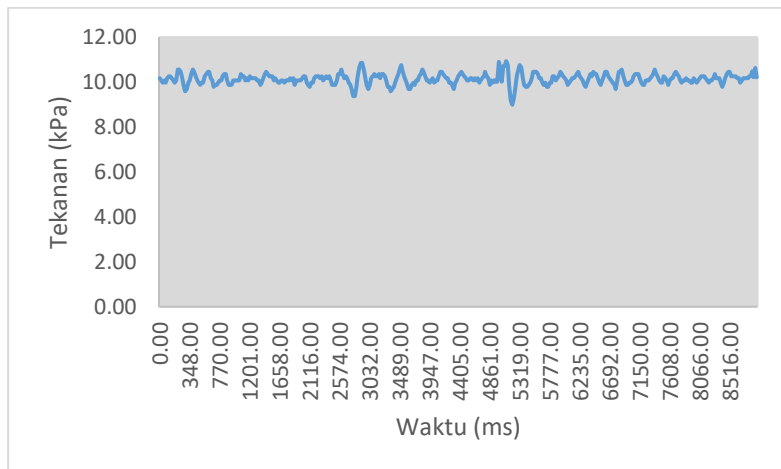
#### 4.5.5. Pengujian Respon Sistem Terhadap Kran Air

Pengujian ini dilakukan untuk melihat respon sistem terhadap perubahan tekanan akibat dari terbuka dan tertutupnya kran air dalam mencapai nilai referensi atau *setpoint*. Pada pengujian ini digunakan nilai  $K_p = 0.000024$ ,  $K_i = 0.000000001$ , dan  $K_d = 0.0009$ . Pengujian pertama dilakukan diawali ketika sistem berada pada *setpoint* dan pengujian kedua dilakukan diawali ketika sistem berada pada kondisi mati.

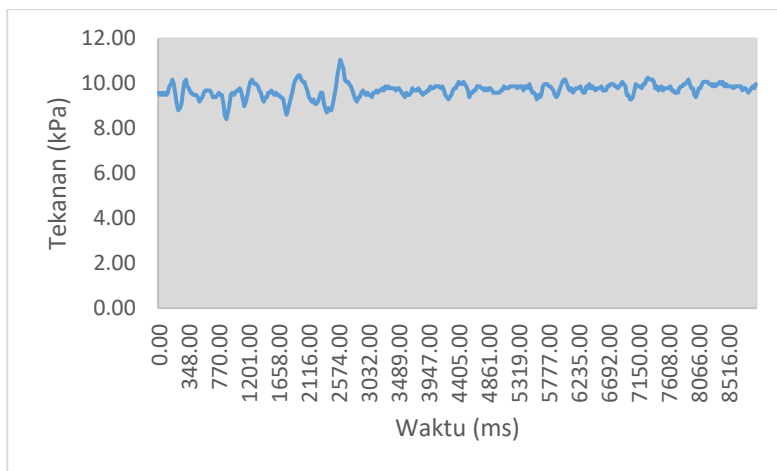
Hasil pengujian pertama ketika kran terbuka dua pada gambar 4.10 didapat sistem tetap stabil, tekanan tetap berada di sekitar *setpoint* dan pompa air tidak bergetar.

Hasil pengujian pertama ketika kran terbuka satu pada gambar 4.11 didapat sistem tetap stabil meski sempat terjadi osilasi sesaat, tekanan tetap berada di sekitar *setpoint* dan pompa air tidak bergetar.

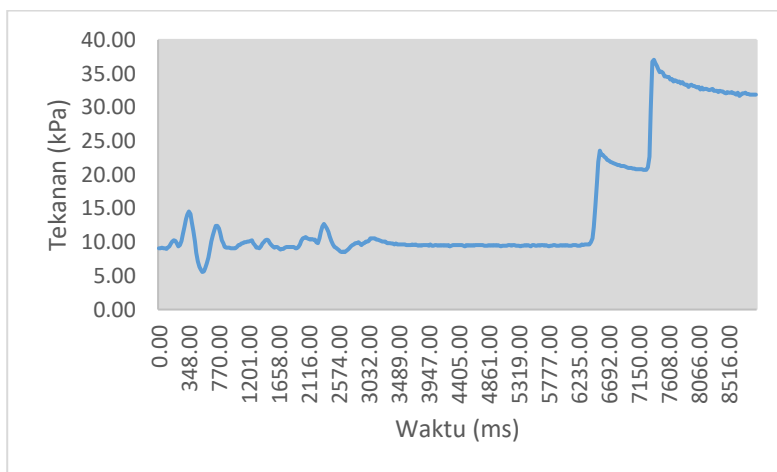
Pada saat semua kran air tertutup pada gambar 4.12 tekanan naik sangat cepat dan membuat pengaman *overpressure* bekerja untuk mematikan pompa.



**Gambar 4.10.** Hasil Pengujian Pertama Ketika Kran Terbuka Dua



**Gambar 4.11.** Hasil Pengujian Pertama Ketika Kran Terbuka Satu



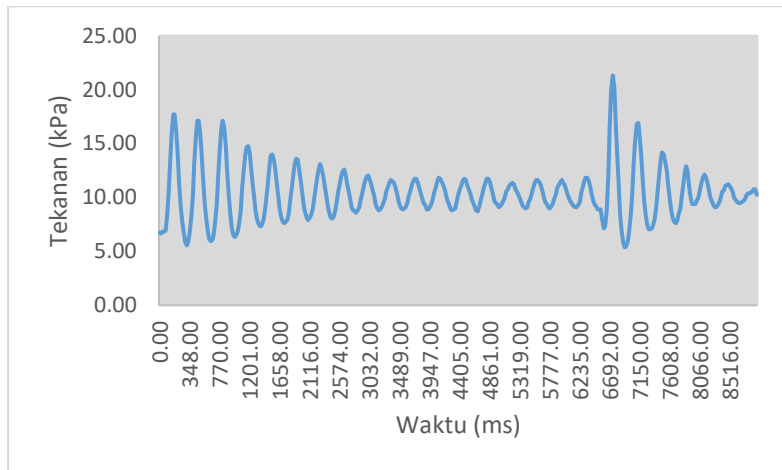
**Gambar 4.12.** Hasil Pengujian Pertama Ketika Semua Kran Tertutup



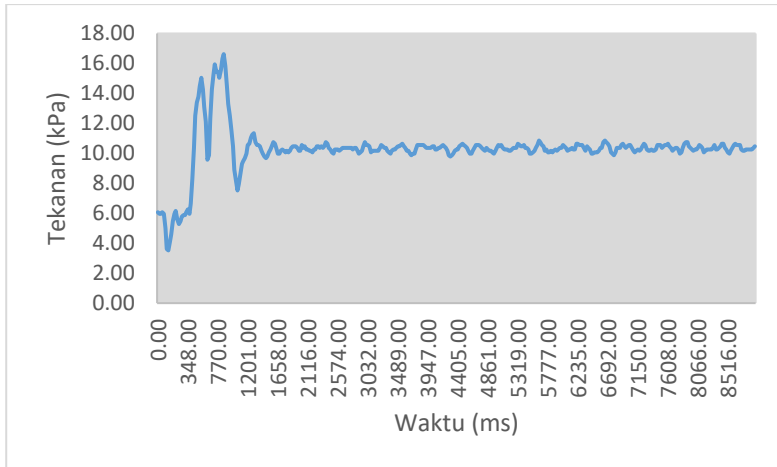
Hasil pengujian kedua ketika kran terbuka satu pada gambar 4.13 didapat sistem mengalami osilasi sesaat, pompa airpun juga bergetar sesaat. Hal ini dikarenakan akibat dari perbedaan level tekanan antara pembacaan tekanan sensor dengan *setpoint* terlalu besar, hal ini mengakibatkan sistem memberikan waktu penyulutan yang sangat lama pada pompa yang mengakibatkan kecepatan pompa sangat pelan. Padahal waktu penyulutan untuk menyalakan pompa air harus lebih cepat dari 6.5 ms.

Hasil pengujian kedua ketika kran terbuka dua pada gambar 4.14 didapat sistem tetap stabil, tekanan tetap berada di sekitar *setpoint* dan pompa air tidak bergetar.

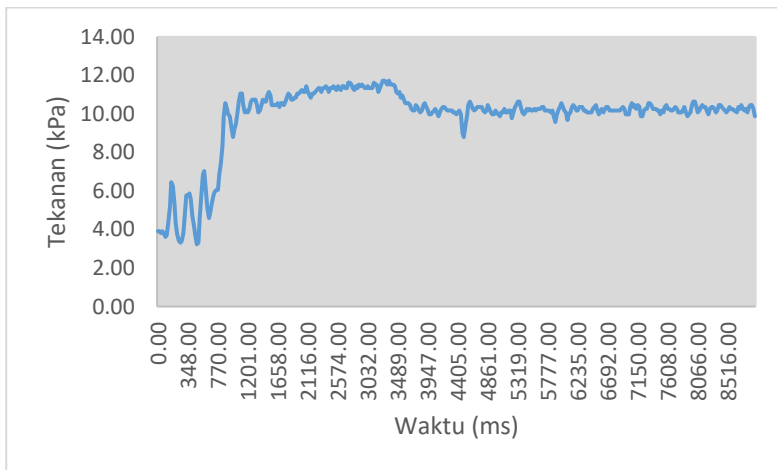
Hasil pengujian kedua ketika kran terbuka dua pada gambar 4.15 didapat sistem tetap stabil, tekanan tetap berada di sekitar *setpoint* dan pompa air tidak bergetar.



**Gambar 4.13.** Hasil Pengujian Kedua Ketika Kran Terbuka Satu



**Gambar 4.14.** Hasil Pengujian Kedua Ketika Kran Terbuka Dua



**Gambar 4.15.** Hasil Pengujian Kedua Ketika Kran Terbuka Tiga

#### 4.6. Pengujian Pengaruh Kontroler Terhadap Debit Air

Pengujian pengaruh kontroler terhadap debit air ini bertujuan untuk mengukur debit air keluaran dari pompa air karena pengaruh kontroler. Dari pengujian tersebut nantinya dapat diketahui apakah kontroler dapat mengendalikan debit air.

Pengujian ini dilakukan dengan cara mengukur debit keluaran pompa air pada satu kran dengan dua kondisi yaitu ketika tanpa menggunakan kontroler dan ketika menggunakan kontroler dengan *setpoint* pada tekanan 11 kPa. Hasil pengujian pengaruh kontroler terhadap debit air tertulis pada tabel 4.7.

**Tabel 4.7.** Hasil Pengujian Pengaruh Kontroler Terhadap Debit Air

Jumlah Kran Terbuka	Debit Air (Lpm)	
	Sistem Ketika Tidak Menggunakan Kontroler	Sistem Ketika Menggunakan Kontroler
3	7,4	8,4
2	11,4	9,6
1	17,6	10,2

Pada tabel 4.7 ditampilkan hasil pengujian pengaruh kontroler terhadap debit air. Dari hasil pengujian didapatkan sistem dapat mengontrol debit air agar tetap stabil pada debit sekitar 8 Lpm dengan rasio perubahan debit kurang dari 2 Lpm.

*Halaman ini sengaja dikosongkan*

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Pada tugas akhir ini telah dibuat sebuah sistem kontrol pompa air yang dapat digunakan untuk mengontrol debit keluaran agar tetap stabil pada ukuran tertentu. Sistem ini menggunakan sensor MPX5100GP untuk mengukur tekanan air pada pipa dan kontroler PID sebagai kontrol sistemnya. Pada sistem ini digunakan tekanan air pada pipa sebagai parameter untuk mengatur kecepatan putaran pompa air untuk mendapatkan debit air keluaran yang stabil.

Ketika kontroler diterapkan pada sistem, debit air keluaran pompa air dapat dikontrol agar tetap stabil pada  $\pm 8$  Lpm dengan rasio perubahan debit kurang dari 2 Lpm ketika *setpoint* tekanan air sistem diatur pada 11 kPa.

Pengaturan Kp, Ki, dan Kd telah menghasilkan respon sistem yang stabil dengan nilai Kp sebesar 0.000024, nilai Ki sebesar 0.000000001 dan nilai Kd sebesar 0.0009 dengan *rise time* lebih kecil dari 1 s dan *settling time* lebih kecil dari 4.2 s dengan rasio kesalahan sebesar 5%.

Kecepatan putaran pompa air diatur menggunakan *variable speed drive* yang komponen utamanya berupa TRIAC. Untuk dapat mengontrol kecepatan putaran pompa air, waktu penyulutan pada *gate* TRIAC haruslah lebih cepat dari 6.5 ms. Ketika lebih lambat dari itu pompa air akan *off*.

#### **5.2. Saran**

Beberapa saran yang penulis bisa berikan untuk pengembangan tugas akhir ini adalah pompa air yang digunakan harus memiliki area kerja yang lebar agar pompa dapat dikontrol hingga kecepatan yang paling rendah sekalipun.

Untuk mendapatkan respon sistem yang lebih baik gunakan tabung peredam tekanan (hidrofor) untuk meredam perubahan tekanan air pada pipa yang terlalu fluktuatif. Dengan meredam perubahan tekanan air pada pipa dapat memperhalus perubahan kecepatan putaran pompa air.

Sistem yang dibuat ini sesuai untuk diaplikasikan pada skala rumah tangga hingga perkantoran.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Amalia, B. Sugiri, A. (2014). “Ketersediaan Air Bersih dan Perubahan Iklim : Studi Krisis Air di Indonesia”. Jurnal Teknik Perencanaan Wilayah Kota Universitas Diponegoro. Vol 3. pp. 295-302.
- [2] Ikponmwosa, O. (2010). “Development of an Electric Water Pump Controller and Level Indicator”. Journal of International Journal of Engineering and Applied Sciences. Vol.3, pp.18-21.
- [3] Chapman, S. (2005). *Electric Machinery Fundamentals 4<sup>th</sup> Edition..* New York : The McGraw, Hill Companies. Inc.
- [4] Deepa, M. (2015). “Design of VFD Drive for a 3-Phase Induction Motor”. Journal of International Journal of Innovative Research in Science. Vol 4. pp.18755-18760.
- [5] Setiawan, R. (2008). Teknik Akuisisi Data. Surabaya : Graha Ilmu
- [6] MCP3008, Microchip Technology , data sheets, 2008.
- [7] Rahman, Z; Ibrahim Ashari, M. 2011. “*Perancangan dan Pembuatan Sistem Proteksi Kebocoran Air Pada Pelanggan PDAM Dengan Menggunakan Selenoid Valve dan Water Pressure Switch Berbasis ATMEGA 8535*”. Institut Teknologi Nasional: Malang
- [8] MPX5100, Freescale Semiconductor, data sheets, 2010.
- [9] Gawande, S. Desmukh, P. (2015). “Raspberry Pi Technology”. Journal of International Journal of Advanced Research in Computer Science and Software Engineering. Vol 5. pp.37-40.
- [10] Raspberry Pi 3 Model B, RS Components, data sheets, 2016.
- [11] Ogata, K. (2010). *Modern Control Engineering 5<sup>th</sup> Edition..* New Jersey : Pearson Education, Inc.

*Halaman ini sengaja dikosongkan*



## LAMPIRAN

### Koding Python Raspberry Pi 3

```
import RPi.GPIO as GPIO
import time
import spidev
import os

# Define GPIO Mapping
PIN_OPTO = 4
PIN_ZCD = 17

# GPIO Setup
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)      # Use BCM GPIO numbers
GPIO.setup(PIN_OPTO, GPIO.OUT) # Optocoupler Trigger
GPIO.setup(PIN_ZCD, GPIO.IN)  # ZCD Interrupt Signal

# SPI Setup
spi = spidev.SpiDev()
spi.open(0,0)

#GPIO.output(PIN_OPTO,True)

def ReadChannel(channel):
    adc = spi.xfer2([1,(8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

def ConvertPressure(data,places):
    pressure = data*100 / float(1023)
    pressure = round(pressure,places)
    return pressure

pressure_channel = 0
```

```

set_point = 11

er = 0 #error
er0 = 0 #error sebelumnya
ser = 0 #sigma error
der = 0 #delta error
tunda = 0 #tunda
tunda0 = 0 #tunda sebelumnya

kp = 0.000024
ki = 0.000000001
kd = 0.0009

GPIO.add_event_detect(PIN_ZCD, GPIO.RISING)

while True :
    pressure_level = ReadChannel(pressure_channel)
    pressure      = pressure_level * 100 / 1023

    er = set_point - pressure
    der = er - er0
    PID = kp * er + ki * ser + kd * der
    ser = ser + er

    if ser > 1000:
        ser = 0
        er0 = er

    if PID > 1000 :
        PID = 1000
    elif PID < -1000 :
        PID = -1000

    tunda = tunda0 - PID * 0.01

    if GPIO.event_detected(PIN_ZCD) :

```

```

if (tunda < 0.009 and tunda > 0.001) :
    time.sleep(tunda)
    GPIO.output(PIN_OPTO,True)
    time.sleep(0.001)
    GPIO.output(PIN_OPTO,False)
elif tunda < 0.001 :
    GPIO.output(PIN_OPTO,True)
else :
    GPIO.output(PIN_OPTO,False)

if pressure > 96:
    tunda = 0.01

if pressure < 4:
    er = 0 #error
    er0 = 0 #error sebelumnya
    ser = 0 #sigma error
    der = 0 #delta error
    tunda = 0 #tunda
    tunda0 = 0 #tunda sebelumnya

tunda0 = tunda

```

*Halaman ini sengaja dikosongkan*

## RIWAYAT HIDUP PENULIS



**Kevin Rosada** lahir di Surabaya pada tanggal 20 Januari 1994. Penulis adalah anak pertama dari tiga bersaudara. Penulis menempuh pendidikan dasar di SD Al-Falah Surabaya pada tahun 2000-2006, kemudian SMPN 12 Surabaya pada tahun 2006-2009, dan lulus dari SMAN 15 Surabaya pada tahun 2012. Pada tahun 2012 juga, penulis diterima sebagai mahasiswa di jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS) dan mengambil bidang studi Elektronika. Semasa kuliah, penulis aktif di Himpunan

Mahasiswa Jurusan sebagai Kepala Departemen LINGPUS HIMATEKTRO ITS 2014-2015 dan juga aktif dalam organisasi pecinta alam Divisi Kalpataru. Penulis dapat dihubungi melalui email [kevinkevros@gmail.com](mailto:kevinkevros@gmail.com)

*Halaman ini sengaja dikosongkan*